

JURDEPÄÄS AINULT TEADMISVAJADUSE
JA JURDEPÄÄSUÕIGUSE ALUSEL

Riigi Infosüsteemi Amet
reg kp 20.12.2011.....
reg nr 1-6/11-6-5-4

AK juurdepääsupirang teatatakse
teavustatud Riit praedirektor
19.11.2011 konkruyaga 1-1-2/11-095.

ASUTUSEISESEKS KASUTAMISEKS

Riigi Infosüsteemi Amet

märge tehtud 20.12.2011 a

Juurdepääsupirang kehtib kuni 20.12.2016 a

Alus: AVTS § 35 lg 1 p 9 ja 10

Lisa 4. Tabel ID-kaardi tarkvara varentteadaolevate vigade kohta, 12.12.2011, Martin Paljak

Nr	Kirjeldus	Tüüp	Seletus	Lahendus	Mõju
1	EstEid aluseks oleva hindamata kaardiplatvormi tehnoloogilise vahetuse riskid on hindamata	risk	Java kaardile üle minnes (jaan.2011) ei hinnatud uue tehnoloogilise keskkonna võimalusi. Otsustati uuel kaardil tarkvaraliselt emuleerida vana kaarti.	On vaja hinnata platvormivahetusest tulenevaid riske, ühtlasi hinnata võimalust kasutada Java kaardil juba olemasolevaid teeke.	Väike või teadmata
2	Algkood ei kompileeru, ei ole Java failistruktuuris	Hea arendus-praktika	Algkood ei ole terviklik, temaga ei kaasne ühtegi buiid skripti, puudu on viited vajalikele lisakomponentidele (Visa OpenPlatform) või sihtplatvormile (leiud 3, 4)	Terviklikuks halduseks tuleb algkoodi säilitada ja hoida koos kõigi sõltuvustega: buiid skriptid, dokumentatsioon tarkvara buidimiseks ja viiteid sõltuvustele.	Suur
3	Versioonikontrolli kasutamine ebaselge	Hea arendus-praktika	Algkood sisaldab suurel hulgal (~19%) välja kommenteeritud koodiridu. See jätab mulje, et algkoodile ei rakendata versioonikontrolli muude vahenditega peale käsitöö. See uudab raskeks muutuste jälgimise rakenduse erinevate versioonide vahel.	Rakendada algkoodile versioonikontrolli. Tulenedes algkoodi omapärale ja võimalikele hoistamisreeglitele, saab kasutada mitte-tsentraalset versioonihaldust (Git, Mercury, Bazaar jne). Git näidis olemas.	Suur
4	Omanduslikud laiendused	Terviklikkus	Rakendus kasutab klasse fr.trustedlogic.* puust, mis ei ole JavaCard 2.2.2 API-s kirjeldatud. Algkoodiga ei kaasnenud ei vastavaid klasse ega algkoodi. Varjatud seotus ühe konkreetse kaardi platvormi tootjaga raskendab vajadusel platvormi Dokumenteerida nõuded. vahetust. Tõenäoliselt seotud leiuga 4.	Otsustada ECDSA allkirjade vajalikkus praeguses rakenduses ja kaaluda avalikule API-le mitte vastava koodi eemaldamist koodipuust.	Suur

ASUTUSEISESEKS KASUTAMISEKS

Märge tehtud: 21.12.2016

Kehtib kuni: 21.12.2021

Alus: AVTS § 35 lg 9,10

Teabevaldaja: Riigi Infosüsteemi Amet

5 Võimalikud omanduslikud laiendused / maagilised konstandid.	Terviklikkus	Algkood püüab luua ECDSA signatuuri objekte kasutades JavaCard 2.2.2 API-s kirjeldamata konstante. Tõenäoliselt seotud leiuga 3.	Kui on vaja kasutada kas kolmandate osapoolte koodi või platvormist sõltuvaid võimalusi, tuleks see dokumenteerida ja võimalusel kasutada nimelisi konstante. Dokumenteerida nõuded ja erisused nii algkoodis kui eraldi seisvalt.	Suur
6 Ebaühtlane koodi külgendus	Koodi loetavus	Algkood on väga ebaühtlase külgendusega ja "kokku pressitud". Muudab raskendatuks koodi lugemise algkujul.	Võtta kasutusele ja dokumenteerida koodi külgendamise standard, mida saab kasutada IDE-s või eraldi seisvate külgendajatega.	Väike
7 Palju kommenteeritud koodi	Koodi loetavus / hea arendus-praktika	Deaktiveeritud koodi hoidmine aktiivse koodi kõrval on segadust tekitav ja võib osutuda vigade allikaks.	Eemaldada kommentaarid, mis sisaldavad algkoodi. Näidis on olemas.	Keskmine
8 Palju kasutuseta koodi	Koodi loetavus / hea arendus-praktika	Algkood sisaldab palju meetodeid ja muutujaid, mida ei kasutata kusagil. Kuigi kompilaator eemaldab üleliigse pagasi kaardile minevast rakendusest, ei ole selguse huvides hea hoida algkoodi hulgas üleliigset.	Eemaldada kasutamata kood. Näidis on olemas.	Keskmine
9 Wildcard importid	Hea arendus-praktika	Heaks tavaks on Java import klauslites viidata ainult kasutatavaid klasse täpse klassi otsiteega.	Viia import klauslid levinud kujule. Näidis on olemas	Väike
10 Extended APDU tugi	Vastuolu programmeerimis-ülesandega / turvalisus	ESTeID klass realiseerib Extended length liidest kuid dokumentatsioon ei kirjelda selle tuge. Dokumentatsioon mainib ühilduvusele viidates suurus 250 ja 254 baiti käsu maksimaalse pikkusena (mis aga ei ole laiendatud APDU)	Viia dokumentatsioon ja rakenduse algkood vastavusse. Vajadusel eemaldada üleliigne kood.	Suur

JURDEPÄÄS AINULT TEADMISVAJADUSE
JA JURDEPÄÄSUÜGUSE ALUSEL

~~ASUTUSESEISEKS KASUTAMISEKS~~
~~Riigi Infosüsteemi Amet~~
~~märge tehtud 20.12.2011 a~~
~~Juurdepääsupirang kehtib kuni 20.12.2016 a~~
~~Alus: AVTS § 35 lg 1 p 9 ja 10~~

11 Väärtuse "NULL" kontroll peale kasutamist	Viga	append_record kasutab SelectedFile muutujat enne kontrolli NULL väärtusega. NULL kontroll ei oma tähtsust	Kui on võimalus, et SelectedFile võib olla NULL, kontrollida seda enne kasutamist. Kui sellist võimalust ei ole ette näha, eemaldada surnud kood. Näidis olemas (Eclipse leid).	Keskmine
12 Klassi meetodite kaitsus	Hea arendus- praktika / turvalisus	Kaks meetodit EstEID ei ole "private" atribuudiga.	Ühtlustada lokaalsete meetodite atribuudid. Näidis olemas	Väike
13 Topeltkontroll samale väärtusele	Viga	internal_authenticate meetod kontrollib APDU P1 väärtuse vastamist nullile kaks korda.	Tõenäoliselt peavad nii P1 kui P2 olema väärtusega 0. FindBugs leid.	Väike
14 Elliptiliste kõverate tugi rakenduses	Vastuolu programmeerimis- ulesandega / turvalisus	EstEID JavaCardi rakendus on loodud emuleerima varajasemat MICARDO põhist EstEID rakendust ja soetud standardeid. Praktikas on kasutuses ainult RSA. Selgusetu on ECDSA toe omanine kaardirakenduses, kui selleks pole (avalikke) vajadusi. Vt ka leiud 1 ja 4	Eemaldada ECDSA-ga seotud kood kuni dokumenteeritud vajaduse tekkeni.	Keskmine
15 JavaCard OwnerPIN-i mitte kasutamine	Vastuolu programmeerimis- ulesandega / turvalisus	PIN kontrolliks kasutatakse kodukootud DES-i põhist süsteemi, kuigi JavaCard pakub OwnerPIN liidest, mida platvorm peaks kaitsma rünnete ulesandega / eest.	Teha otsus, kas dokumenteerida JavaCard-i pakutavate võimaluste mitte kasutamine või võtta need kasutusele.	Suur
16 Rakenduse võimete mittevastamine nõuetele	Vastuolu programmeerimis- ulesandega / turvalisus	Eesti ID-kaart (2011) versioon ei toeta sertifikaatide uuendamist ega võtmete uuesti genereerimist. Ometigi on rakenduses tugi kuni 8 võtmepaari jaoks. Seotud keerukus lisavõtmete haldusega muudab rakendust vajalikust keerukamaks	Viia dokumentatsioon ja rakenduse algkood vastavusse. Vajadusel eemaldada üleiligne kood.	Väike

17	Ebaselge objektide "ristviitamine" liiga suurtes massiivides O, G_byte ning G_i	Koodi loetavus / hea arendus-praktika	Massiivi O kasutatakse ainult kahe viite (SelectedFile ja RunningSignature) hoidmiseks, mälu aga küsitakse 22 viite jaoks. Edasi, massiivi G_byte kasutatakse ainult kuue elemendi hoidmiseks, mälu aga küsitakse 11 elemendi jaoks. Lõpuks, massiivi G_i kasutatakse 1 elemendi hoidmiseks, mälu aga küsitakse 10 elemendi tarbeks.	Mälu tuleb küsida nii mitme elemendi jaoks, kui tegelikult vaja (2/6/1 vs 22/11/10) Lihtsam ja loetavam oleks kasutada eraldi muutujaid vastavate viidete hoidmiseks.	Väike
18	Mutujate G_byte ning G_i hoidmine EEPROM-is	Hea arendus-praktika / töökindlus	Massiiv G_byte võib teatud juhtudel (olenevalt kaardi platvormist, sõltuvalt kaardil olevast mälust) olla hoitud mitte RAM-is vaid EEPROM-is. EEPROM-i sisu on loetav piisava huvi korral. EEPROM "kulub" kasutades. Massiiv G_i võib teatud juhtudel (olenevalt kaardi platvormist, sõltuvalt kaardil olevast mälust) sattuda mitte RAM-i vaid EEPROM-i sisu on piisava huvi korral loetav	Optimeerida massiivi suurus. Valida kaardi platvorm vastavalt nõuetele või kohandada/optimeerida rakendust vastavalt kaardi tegelikele võimetele.	Keskmine
19	PIN koodi sisaldava muutuja G_i hoidmine viisil, mille puhul pole välistatud selle sattumine avalikku mäluosasse.	Turvalisus		Tundlikku infot sisaldavaid muutujaid tuleb käsitleda vastutustundlikul viisil	Suur
20	Ebaselge objektide "ristviitamine" liiga suures massiivis G_bool	Koodi loetavus / turvalisus	Massiivi G_i kasutatakse ainult kolme elemendi hoidmiseks (PIN koodide kontrollitud olek), mälu aga küsitakse 5 elemendi jaoks. Kriitilist turvainfot sisaldavaid muutujaid tuleks koodis käsitleda äärmise selgusega.	Kui kasutatakse ainult kolme elemendi, küsida mälu kolme elemendi jaoks. Lihtsam ja loetavam oleks kasutada eraldi muutujaid või eraldi OwnerPIN objekte.	Väike
21	G_bool sisse viitamine nii nimedega kui	Koodi loetavus / turvalisus	Kasutusel on nii konstandid (bo_PIN1) kui numbrilised konstandid (verify_pin). Kriitilist turvainfot sisaldavaid muutujaid tuleks koodis	Selguse ja ühtluse ja vigade vältimise tõttu kasutada alati nimelisi konstante	Väike

konstantidega		käsitleda äärmise selgusega.	
22 Ebaühtlane nimede kasutus koodis	Koodi loetavus / hea arendus-praktika	Meetodite ja muutujate nimed ei ole ühtlased, kasutusel on nii CamelCase kui Camel_Case kui camel_case	Ühtlustada nimed vältimaks vigu ja koodi Väike ettearvamatut käitumist.
23 G_Scratch muutuja EEPROM-is hoidmine praktika	Hea arendus-praktika	Massiiv G_Scratch võib teatud juhtudel (olenevalt kaardi platvormist, sõltuvalt kaardil olevast mälust) olla hoitud mitte RAM-is vaid EEPROM-is. EEPROM-i sisu on loetav piisava huvi korral.EEPROM "kulub" kasutades.	Optimeerida massiivi suurus. Valida kaardi platvorm vastavalt nõuetele või kohandada/optimeerida rakendust vastavalt kaardi tegelikele võimetele.
24 G_Sec_Mess_Array muutuja EEPROM-is hoidmine	Hea arendus-praktika	Massiiv G_Sec_Mess_Array võib teatud juhtudel (olenevalt kaardi platvormist, sõltuvalt kaardil olevast mälust) olla hoitud mitte RAM-is vaid EEPROM-is. EEPROM-i sisu on loetav piisava huvi korral.EEPROM "kulub" kasutades.	Optimeerida massiivi suurus. Valida kaardi platvorm vastavalt nõuetele või kohandada/optimeerida rakendust vastavalt kaardi tegelikele võimetele.
25 G_Chaining_Buffer muutuja EEPROM-is hoidmine	Hea arendus-praktika	Massiiv G_Chaining_Buffer võib teatud juhtudel (olenevalt kaardi platvormist, sõltuvalt kaardil olevast mälust) olla hoitud mitte RAM-is vaid EEPROM-is. EEPROM-i sisu on loetav piisava huvi korral.EEPROM "kulub" kasutades.	Optimeerida massiivi suurus. Valida kaardi platvorm vastavalt nõuetele või kohandada/optimeerida rakendust vastavalt kaardi tegelikele võimetele.
26 0x6B00 vs 0x6A86	Vastuolu / Viga	Micardo kasutab enamikel juhtudel 0x6A86 veateadet sobimatute P1/P2 väärtuste puhul (ISO7816.SW_INCORRECT_P1P2), kui rakendus kasutab ISO7816.SW_WRONG_P1P2 (0x6B00)	Dokumenteerida kaardi liides. Viia kood Keskmise vastavusele dokumentatsioonis ja MICARDO rakendusele.

27	Sümboolsete veakoodide kasutamine (kokku 11 säästast koodi)	Koodi loetavus / hea arendus- praktika	ISO7816 klassis olevaid Status Word (SW) koode Asendada numbrilised konstandid tuleks kasutada loetavuse huvides numbriliste konstantide asemel. Olemasolevatele maagilistele konstantidele anda nimed. Kehtib veakoodide 0x6401, 0x6700, 0x6884, 0x6983, 0x6988, 0x6998, 0x6A86, 0x6A88, 0x6A89, 0x6F01 ja 0x6F02 puhul	Keskmine ISO7816 klassi nimeliste konstantidega, kirjeldada ja dokumenteerida ülejäänud.
28	Puuduv dokumentatsioon	Koodi loetavus	Kui eemaldada algkoodist versioonihalduse eesmärgil väljakommenteeritud kood, jääb inimesele lugemiseks (mis on kommenteerimise TEGELIK eesmärk) mõeldud kommentaariridu järgi ainult 25 rida, ehk alla 1% koodist.	JavaCard on keerukas ja limiteeritud platvorm, kuid seda ainult kaardil. Võttes arvesse rakenduse pikaajalisust (5 aastat muutmata kujul) ja olulisust, dokumenteerida algkood nii JavaDoc'i tasemel, kui ka meetodite sees. Eriti pöörata tähelepanu (kordus)kasutatavatele andmestruktuuridele (APDU puhvri (GB) tagumine osa) Vähem koodi on loetavam ja hallatavam kood, koondada APDU chaining kontroll ühte kohta enne ülejäänud process() loogikat.
29	APDU Chaining tugi ja kontroll	Koodi loetavus	APDU chaining on toetatud ainult dekrüpteerimiseks kuid kontroll tehakse samamoodi (coppypaste) kõigi ülejäänud käskude puhul.	Lubada ainult ette nähtud operatsioonid
30	Dekrüpteerimine signeerimisvõtmega	Vastuolu	ESTEID MICARDO rakenduse spetsifikatsioon näeb ette signeerimisvõtmega ainult digiallkirjade arutamist (punkt 4.2.4), mis praktikas on sha1WithRSAEncryption, kuigi Turvakeskkondade ja operatsioonide risttabelis on "dekrüpteerimine PIN kontrolliga" lubatud ka allkirjavõtmele.	Keskmine allkirjavõtmega, ehk siis signeerimine/krüpteerimine.

JURDEPÄÄS AINULT TEADMISVAJADUSE
JA JURDEPÄÄSUÜGUSE ALUSEL

ASUTUSESISESEKS KASUTAMISEKS
Riigi Infosüsteemi Amet
määrge tehtud 20.12.2011 a
Juurdepääsupiirang kehtib kuni 20.12.2016 a
Alus: AVTS § 35 lg 1 p 9 ja 10

31	Aplikatsiooni AID küsimise käsule (kaardis v 3.0) antakse ebakorrektne vastus 0x9000	Vastuolu / disainiviga (varasemalt teada)	Ainuke aplikatsioon EstiEID kaardil on ESTEID ise. Vaid selle küsimuse käsule tohib vastata 0x9000. Hetkel näiteks Windows 7 peab ID-kaarti vääralt "NIST SP 800-73 I[PIV] tüüpi autentimisvahendiks.	Kaart peab end ja omi ressursse identifitseerima korrektselt, et mitte põhjustada häireid teiste kiipkaartide kasutamises.	Väike
32	Aplikatsioon (v 3.0) reageerib dokumenteerimata APDU' dele. MacOS/X all toob see kaasa kaardi PUK koodi lukustumise	Vastuolu / disainiviga (varasemalt teada)	Pärast seda kui MacOS/X leiab 31 mõjul hõivab kaardi ja saadab prooviks CHV0 (80:20:00:00 APDU -ISO VERIFY with a proprietary CLA byte), kaart vähendab PIN-loendurit ja vastab - "incorrect reference data" (0x6984), mille peale MacOS proovib uuesti kuniks kaart lukustub.	Kaart peab käituma korrektselt, et mitte põhjustada lukustumist ja halba kasutajakogemust	Keskmine
33	Aplikatsioon v 3.0 vastab vale pikkusega PINi sisestamisele vääralt	Vastuolu / disainiviga (varasemalt teada)	Kaart peaks PINi vale pikkuse korral vastama 0x63CX ("incorrect data in command field"), kuid vastab 0x6984.	Kaart peab käituma vastavuses dokumentatsiooniga ning harmoonias baastarkvaraga	Väike
34	Väära PINi sisestamisele reageerib aplikatsioon vääralt	Vastuolu / disainiviga (varasemalt teada)	Kaart peaks väära PINi korral vastama 0x63CX kuid vastab 0x630X. See eksitab enamikku standardtarkvarast.	Kaart peab käituma vastavuses dokumentatsiooniga ning harmoonias baastarkvaraga	Väike
35	Aplikatsioon võimaldab PINi "vahetust" endiseks.	turvaviga (varasemalt teada)	Erinevalt eelmistest versioonidest, PIN vahetamisel või PUK-koodiga PINi lahtilukustamisel on võimalik taas panna vana PIN sellega.	Turvamudel peaks olema kirjeldatud ja aplikatsioon peaks käituma vastavuses	Väike

JUURDEPÄÄS AINULT TEADMISVAJADUSE
JA JUURDEPÄÄSUÜGIGUSE ALUSEL

~~ASUTUSESISESEKS KASUTAMISEKS~~
~~Riigi Infosüsteemi Amet~~
~~20.12.2011 a~~
~~20.12.2016 a~~
~~Alus: AVTS § 35 lg 1 p 9 ja 10~~

	Väike	
36 Applikatsioon vastab varasemalt käsule "SELECT FILE" kahekordse 0x62 FCI	Kaart vastab käsule "SELECT FILE" kahekordse 0x62 FCI sektsiooniga. See käitumine on hetkel parandatud "nõpitanisega" baastarkvaras.	Parandada vale käitumine kaardiapplikatsioonis.
37 Applikatsioon väljastab väärteate 0x6985, mis ei võimalda turvamudelilt täies mahus realiseerida.	Hea turvapraktika kohaselt peaks kaart valed PIN'i sisestusele vastama 0x6982 "Security status not satisfied", kuid vastab 0x6985 "Conditions of use not satisfied" (justkui oleks tegu mingi väikese veaga). Tagajärg: PIN1 puhverdamise kood (baastarkvaras?) ei tööta korrektselt.	Turvamudel peaks olema kirjeldatud ja aplikatsioon peaks käituma vastavuses sellega.
38 Isikuandmete faili lõpp on polsterdatud tühikutega	Hetkel baastarkvara on sunnitud spetsiaalselt parsima ja eemaldama isikuandmete faili lõpust tühikud.	Viia isikuandmete faili (3F00/EEEE/5004) kirjed vastavusse standardiga.
		Keskmine